N89-25162

TRUSS: AN INTELLIGENT DESIGN SYSTEM FOR AIRCRAFT WINGS

Preston R. Bates* and
Daniel P. Schrage
Georgia Institute of Technology
Atlanta, GA

* Graduate Research Assistant

Abstract

Competitive leadership in the international marketplace, superiority in national defense, excellence in productivity, and safety of both private and public systems are all national defense goals which are dependent on superior engineering design. In recent years, it has become more evident that early design decisions are critical, and when only based on performance often result in products which are too expensive, hard to manufacture, or unsupportable. Better use of computer-aided design tools and information-based technologies is required to produce better quality U.S. products. This paper outlines a program to explore the use of knowledge based expert systems coupled with numerical optimization, database management techniques, and designer interface methods in a networked design environment to improve and assess design changes due to changing emphasis or requirements. The initial structural design of a tiltrotor aircraft wing is used as a representative example to demonstrate the approach being followed.

Introduction

As it becomes more evident that the early stages of design of complex products are where critical life cycle decisions are made, there is increasing pressure to obtain more knowledge and address more requirements early. For advanced aeronautical vehicles, this requirement growth is depicted in Figure 1. The relationship between design freedom and knowledge is illustrated in Figure 2. The obvious goal is to steepen the knowledge curve early to take advantage of the design freedom.

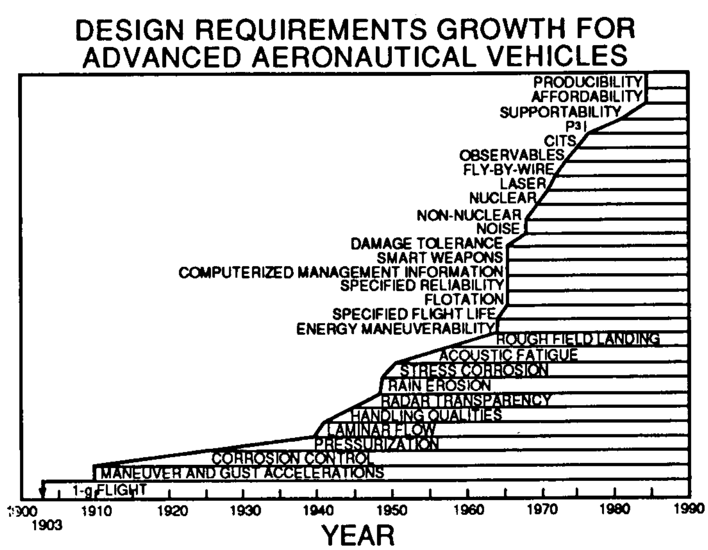


Figure 1

DESIGN FREEDOM AND KNOWLEDGE

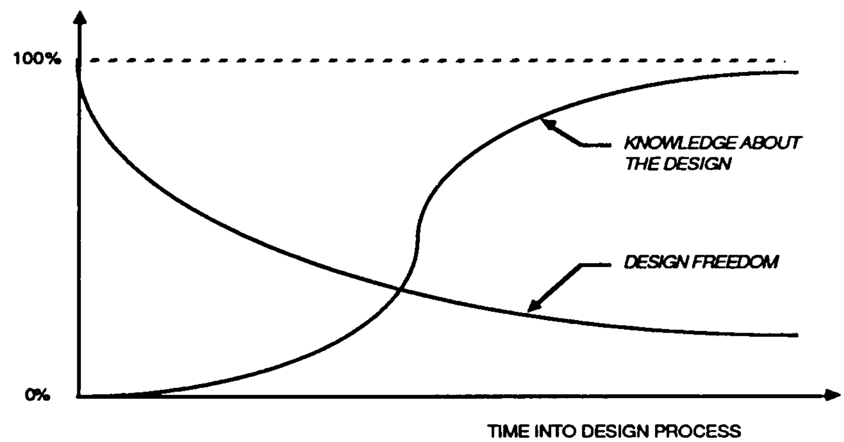


Figure 2

While this approach has received much emphasis, and terms like "design for producibility" and "design for supportability" have become popular, the implementation has been far harder to achieve. One reason is that few design engineers know how to interpret producibility, supportability, etc. requirements, and few manufacturing engineers and logisticians know much about design. There is a movement to correct this deficiency as discussed in References 1 and 2, and DoD programs have been established such as Unified Life Cycle Engineering (ULCE), Reliability and Maintainability in Computer Aided Design (RAMCAD), and Concurrent Engineering.

Necessary research areas have also been identified in Reference 3. Two major sub-areas of design theory and methodology have been identified along with three supporting disciplines whose development is critical to the future growth of the design field. The two major sub-areas are: Conceptual Design and Innovation, and Quantitative and Systematic Methods. The three major supporting disciplines are: Intelligent and Knowledge-Based Systems, Information Integration and Management, and Human Interface Aspects in Design. At the Georgia Institute of Technology, the School of Aerospace Engineering recognized the need for these research areas and supporting disciplines several years ago and has initiated a Laboratory for Information Technology in Engineering (LITE) to address them.

The LITE Program

LITE is a multidiscipline, multi-school effort whose key players stem from the Schools of Aerospace, Mechanical, and Industrial Engineering, as well as the Artificial Intelligence Branch of the Georgia Tech Research Institute (GTRI). The approach taken by the LITE program is to place design information and knowledge at the center of an integrated design process (Figure 3). The LITE philosophy is built upon three key aspects to improve the design process. Primarily investigated by the aerospace school, the first aspect is design decision-making and analysis, addressing synthesis, parametric design, and the use of artificial intelligence (AI) technology in them. Second is information integration and management through the application of shared databases (relational and object-oriented) that fulfill the unique requirements of engineering CAD systems, explored chiefly by the mechanical engineering school. Examined principally by the industrial engineering school, the third is human interface aspects in design, which studies the overall impact of integrated design technology upon individual designers and their organizations.

The LITE design study will focus on issues relative to tiltrotor aircraft like Bell Helicopter's very successful XV-15 shown in Figure 4. Over the past two years, Georgia Tech has been developing the necessary tiltrotor expertise, the design analysis tools, and the interfaces with the tiltrotor industry and government. The V-22 "Osprey" is in full-scale engineering development and will eventually provide approximately 1100 aircraft for the Marine Corps, Navy, Air Force, and Army. In addition, NASA, the FAA, Airport Authorities, and industry are all investigating the use of commercial tilt rotors to relieve airport congestion and improve regional airline productivity. The Europeans have also initiated their own commercial tiltrotor development program, known as EUROFAR, providing the element of foreign competition as well.

INTEGRATED DESIGN PROCESS

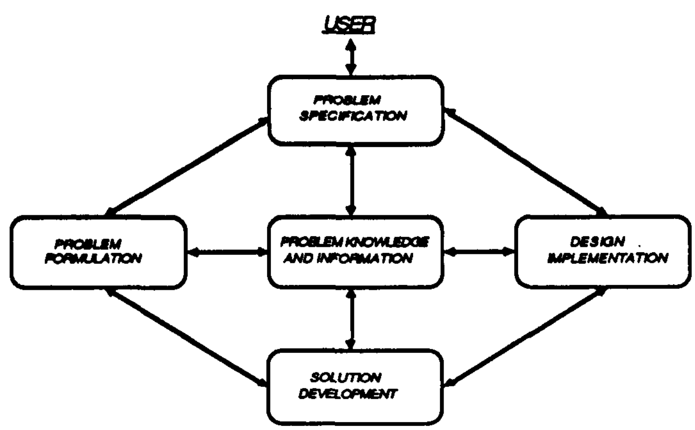


Figure 3

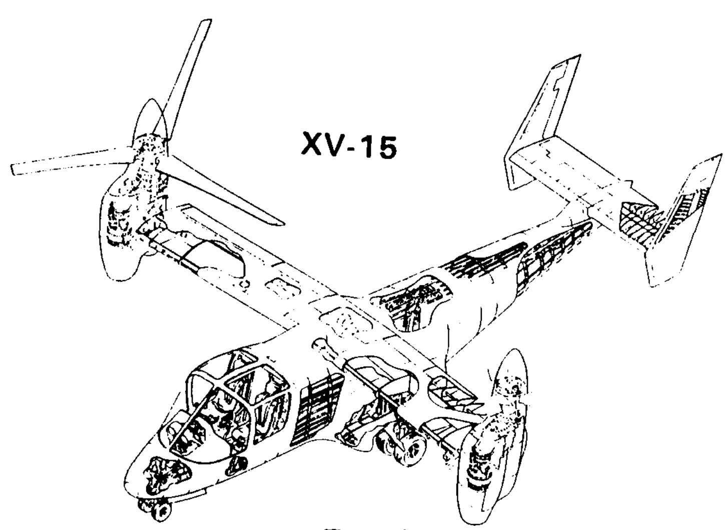


Figure 4

Specifically addressed is the structural design of the tiltrotor wing, a sufficiently intricate aircraft subsystem encompassing a wide variety of functional criteria, aeroelastic, manufacturing, and supportability issues. It presents a complexity that requires a hierarchical problem breakdown into multiple levels of the tilt rotor system, e.g. aircraft, wing, wing box, spar, spar cap. Additionally, it provides a framework to analyze how aircraft design requirements and parameters are related in civilian vs. military vs. research missions. For example, commercial designs are driven by producibility, such as minimum cost per seat-mile, compared to low cost and risk for a research aircraft. In a military application, the design is strongly impacted by operations and support issues including structural inspection and tracking, battle damage repair, and the need to operate in austere or shipboard environments.

The typical wing structural design process consists of the conceptual, preliminary, and detail design phases. At the conceptual level, general parameters related to the entire aircraft are specified. Definition of wing area, thickness to chord ratio, span, sweep, and fuel volume requirements result in geometric constraints governing the internal wing structure. During preliminary design, attempts are made to find the best way to "put the bones in the meat" of the aircraft by laying out major structural components that satisfy these constraints. At this stage, trade-off studies, coupled with mathematical analysis and optimization, are performed on the various structural configurations. Consideration of producibility, maintainability, and supportability are crucial at this level. Finally, at the detail phase, the wing subcomponents (panels, ribs, spars, etc.) are considered individually, resulting in shop drawings for their manufacture.

TRUSS (Tilt Rotor Unified Structural design System) is the present focus of LITE that incorporates in its development all of the research areas and supporting disciplines discussed so far. TRUSS is an integrated design system that attempts to automate the basic wing structural design process shown in Figure 5. From the figure it is easy to see that opportunity for automation exists by applying state of the art technologies in artificial intelligence, database management techniques, interactive geometric modelling, finite element analysis, and optimization techniques. Such a project requires a team effort, and at present there are five graduate students from the participating schools concentrating on these individual areas.

TRUSS will generate potential structural configurations commonly incorporated by industry, optimize them on a first level to meet geometric and other constraints, and finally evaluate the feasible concepts according to some common criteria, such as minimum cost or weight. Subsequently, detail design can begin at the subcomponent level. Shown in Figure 6 is an example structural decomposition of a tiltrotor aircraft. During the integration of the aircraft, only one structure from each area (fuselage, wing, etc.) may be chosen. One particular goal of TRUSS is to effectively track the reasons and decisions for these choices, at least at the wing level. Such decision tracking provides potential payoffs in product cost and time to design, and would be applicable to other areas of the aircraft as well.

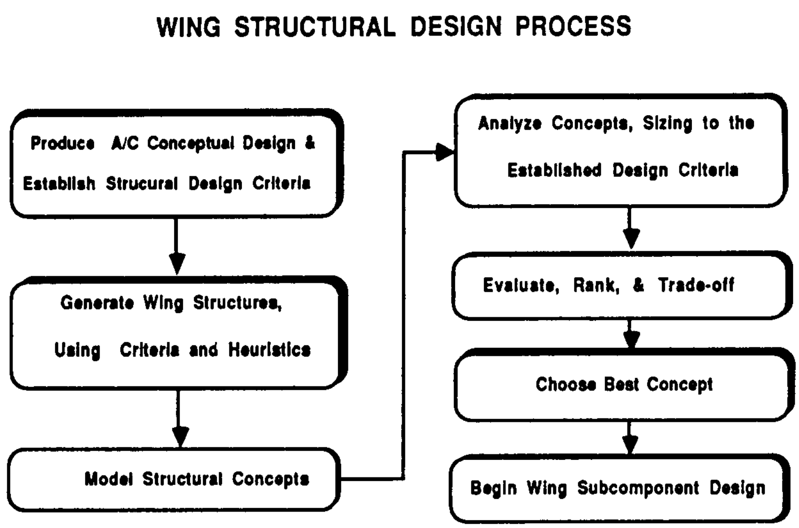
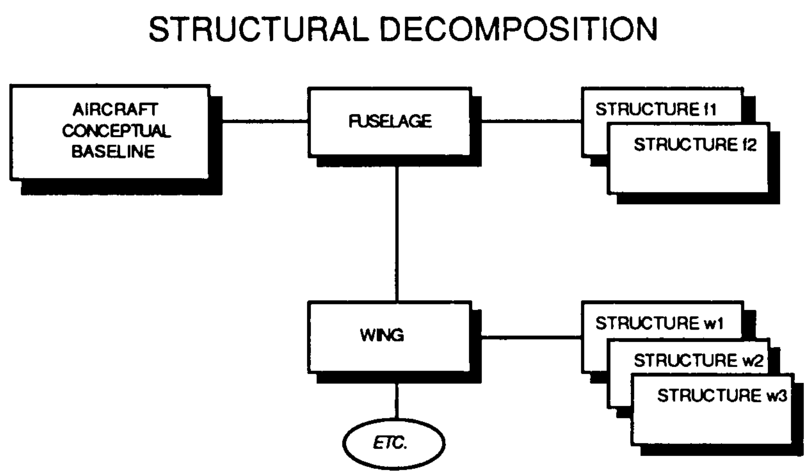


Figure 5



Effective Decision Tracking Will Provide Improvements In:

- PRODUCT COST
- TIME TO DESIGN
- APPLICATIONS TO OTHER SYSTEMS

Figure 6

Description of TRUSS

The issues and details of TRUSS and its various components will now be discussed in some depth. The status of their development and researchers involved will be described. Finally, an overview of the long term research objectives will be presented.

System Architecture

The architecture chosen for the system is an executive-centered concept shown in Figure 7. The executive acts as the prime communicator between all other parts of the system, including the user, and is responsible for the correct order of execution and data flow between them. Reference 4 describes a successful application of this quasi-procedural approach in detail. Individual procedural modules written in various traditional languages are linked together at runtime in an appropriate order that permits computation of a requested design variable. Part of the executive known as the computational path generator uses information on the required inputs and outputs of the available procedural modules, performs a heuristic search of a tree structure relating the design variables, and decides on the order of the routines to be executed.

This layout offers a lot of flexibility when an upgraded program is substituted into the system, requiring only one new interface to the executive. As a result, new modules can easily be added to the executive as the level of required design detail increases. From a database standpoint, multiple databases may be connected to the executive, utilizing the best features of both relational and object-centered management techniques.

TRUSS ARCHITECTURE

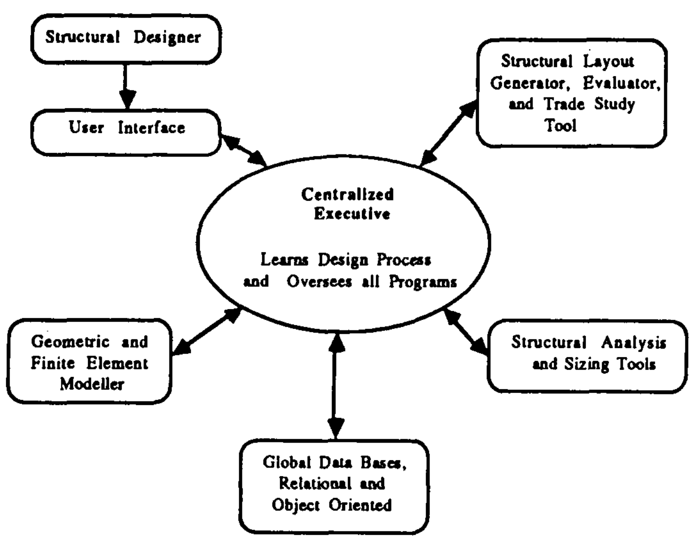


Figure 7

There are, however, some disadvantages of this architecture. Placing all of the responsibility on the executive slows down the transfer of information and requires more working memory, resulting in longer computing times. Some solutions to this include highly efficient database management procedures and the use of parallel processing. To quicken the research, LITE members will prototype TRUSS modules separately on different computers, resolving their own errors before trying to integrate the whole system.

The executive centered system was selected over a global database centered concept. Although a central database architecture may provide for faster computing time, its inherent dependency on specific software requires specialized interfaces between the database and other modules. When improved software is inserted into this system, multiple interfaces must be developed, causing increased system downtime and costs.

Problem Definition

Figure 8 presents the tiltrotor wing problem scope addressed by TRUSS. On the topmost levels, all aircraft wings are members of broadly defined groups. Tiltrotor wings are located under the V/STOL transport category. Further classification breaks down into mission application, functional discipline (structures, aerodynamics, aeroelasticity, flight controls, power train, etc.), and major structural materials. Next is the subsystem level (wing box, leading edge, fixed trailing edge, etc.), and finally the component level where actual pieces of structure, such as skin, ribs, and spars, can be found. These may even be further broken down into their components as well, such as web, chord, stiffener, etc. It is beyond the scope of TRUSS to address the entire problem space, so one of the branches in the hierarchy will be chosen as the initial design task. When all of the TRUSS participants have agreed upon the proper problem definition, module prototyping will begin. For example, initial development work might be identified by the shaded boxes, which indicate the material selection and structural analysis of the wing box spar in a commercial aircraft.

Proper design of TRUSS requires a detailed understanding of the tiltrotor wing structural design process, as well as a knowledge of the proper analysis tools and available design technologies. Figure 9 represents a design network of the required tasks, forms of data, and task interactions encountered in the design sequence. Once these have been identified, LITE team members construct a flowchart which places design tools, current or required, in their proper places in the sequence (Figure 10). Also represented on the flowchart are the state-of-the-art technologies required for design automation, such as artificial intelligence, interactive graphics facilities, and optimization. Such a view of the design process is not cast in concrete, and must be continuously re-examined and updated. In this manner, the causes of problems in the current design process can be pinpointed and reasonable solutions for improvements made. Interfacing issues among the different design tasks may also be identified. Once finalized into a valid form, such a flowchart can be programmed as a script into the centralized executive of TRUSS which would oversee all data interactions between the modules and the user.

TILTROTOR WING PROBLEM SCOPE

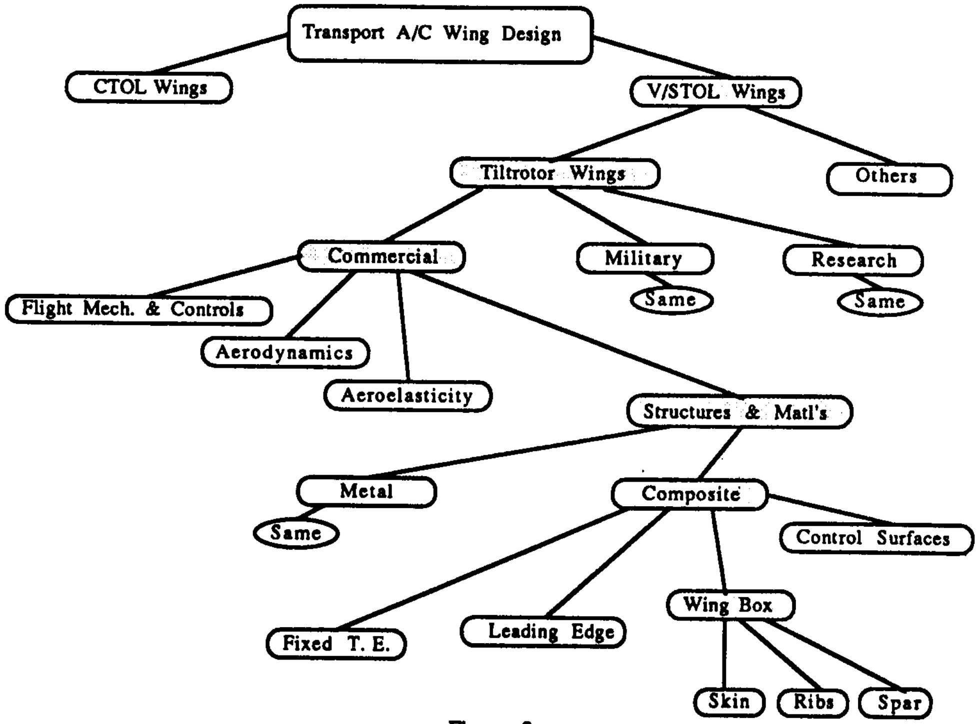


Figure 8

DESIGN NETWORK

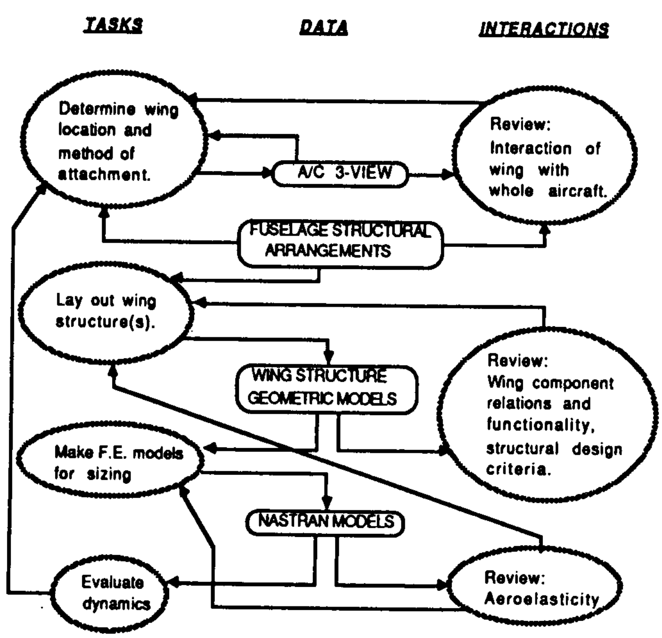


Figure 9

TILTROTOR WING STRUCTURAL DESIGN PROCESS

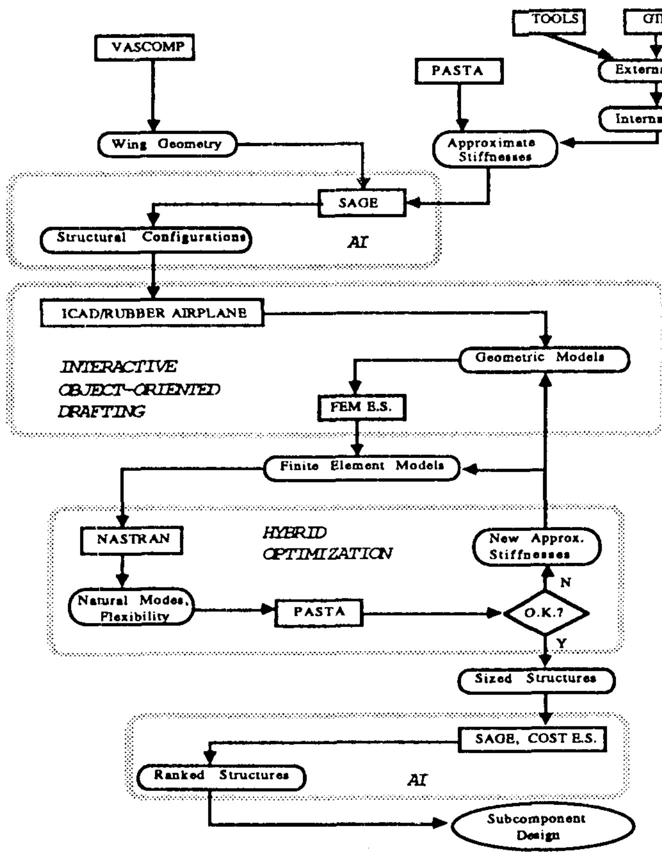


Figure 10

<u>User Needs and Interface</u>

One of the most important modules of the TRUSS system is the user interface. A poor interface results in a poor system. Representative designers from industry will play a key role in the development of this module by evaluating prototype interfaces and offering suggestions and requirements for improvements. A number of user needs have already been identified.

First, the system must be able to support the needs of a number of different departments within an organization. For instance, structural design reviews, marketing reviews, manufacturing consultations, and a variety of other management/administrative tasks are crucial to design decision support. While satisfying the structural designer's needs for adequate technical depth, the system should also provide broader information for upper level organizational requirements within the framework of project time and cost constraints.

Next, the system should have the ability to learn from interactive sessions with the user by remembering exactly how the user created the structural design. As a result, when minor modifications to the design must be made, the user need only change a couple of parametric values and the system automatically updates the design. This is also known as parameter-based design vs. geometry-based. Boeing

Commercial Airplane Company has successfully incorporated this feature on their wing configuration design system, CDCS, Reference 5. Still recognizing the advantages of geometric design, TRUSS will also have a drafting facility.

Finally, the system should present information to the designer in the most effective way for him to use the design data. This entails the use of user defined pop-up windows and access to the rule base, knowledge base, and databases. A sample user interface concept for TRUSS that reflects these criteria is shown by Figure 11.

TRUSS USER INTERFACE

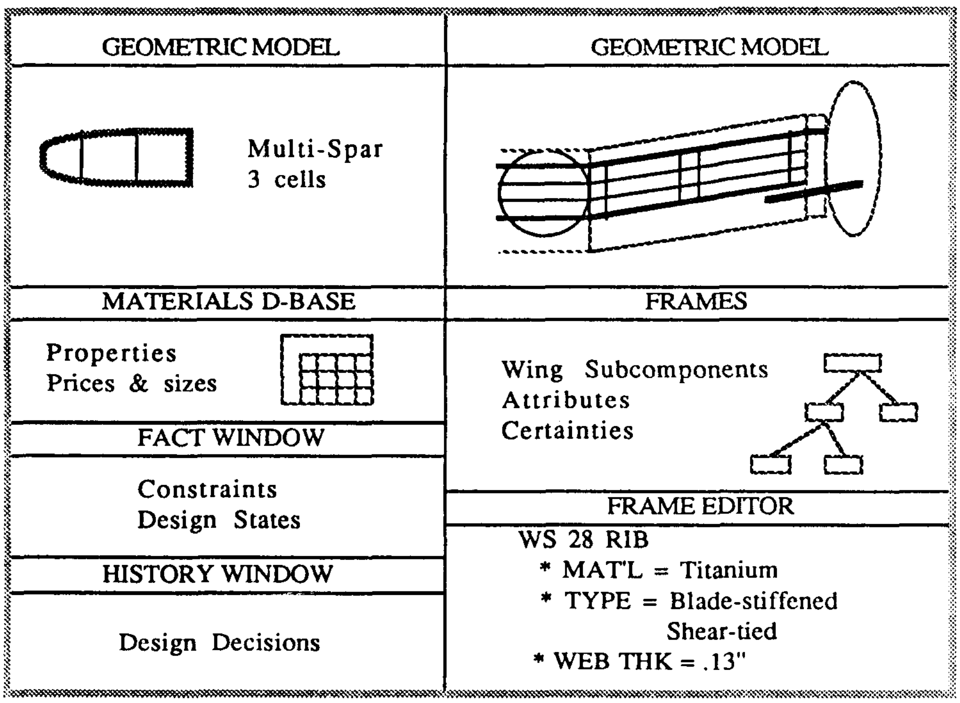


Figure 11

Database Issues

Databases are imperative to the proposed design system. They must serve as the repository for all information describing different design versions, and all design decisions up to any given time. All other TRUSS modules are dependent on this information to perform their required tasks. Specific data applications include material specifications and properties, geometric modelling data, meta-level and parametric descriptions of feasible structural configurations, sensitivity data, and machining specifications.

Reference 6 provides some detailed issues about the function of the database in TRUSS, some of which are discussed in the following paragraphs. Currently, one LITE team member is focusing his research specifically on the implementation of these items.

Modeling of complex design objects: Engineering CAD databases have requirements above and beyond those used for business applications. Engineering design problems involve large systems with complex interactions among their subsystems and components. So the need arises for more advanced modelling techniques. The use of object-oriented databases with sophisticated data structures such as semantic networks, in addition to traditional relational structures, is required for these modelling tasks.

Version/alternative control: There is seldom any single solution to a design problem, with the possibility of several optimal solutions existing that all satisfy the product requirements and constraints. Hence arises the need for the storage and arrangement of data specific to these different versions or alternatives, which may be called upon by other application programs.

View/configuration management: A system like TRUSS involves many users with different needs for design data. Even the separate modules of the system have different requirements for the same data. These differing 'views' of the design object must be supported. On one hand is an administrative view of the data where a project manager might be interested only in a few key parameters, such as system cost or production time estimates. On the other hand, the design engineer may need to know very detailed information like individual structural component weights, their raw material types, and associated costs.

Dynamic schema definition: A database schema defines the objects modelled in the database and their binding relationships. For multiple views, these schemas must be flexible for modification and extension, preferably without reloading the data base or recompiling the schemas, thus working dynamically with the system.

Concurrent control: A unified design system involves the communication among several of its modules, and among several users in a networked workstation environment. The ability to control the data of several parallel processing applications is desired from the database manager to ensure data integrity along with faster computing times. In AI terminology, such a 'blackboard' displays and modifies all data between separate application programs acting in parallel.

Partial integrity and constraints: Design of a particular object is done in an iterative fashion. While many types of design have a preferred sequence of activities that take place, the database system should not impose constraints on this sequence. It may be necessary to allow and manage inconsistent database states.

Management utilities: These include the basic tasks of data backup, recovery, security, operational accounting and performance statistics, and off-line storage and data archival.

Hardware and Software

Significant chokepoints arise in the current design process whenever hardware or software incompatibilities exist. Reference 7 tells that such days for engineers are coming to an end. Apollo Computer, Inc. has developed a set of products which create a heterogenous networked computer environment using the best features of different hardware. The requirements of TRUSS point toward the use of this type of design environment. To address this issue, LITE members are compiling a comprehensive listing of available programs used by industry and government organizations, the hardware they run on, and the language the source code is written in. Especially challenging is the marriage of traditional programs used in a "number crunching" environment to new AI tools and machines. The wide variety of such combinations is shown in Figure 12. The proper choice of hardware and software combinations is a primary goal of the TRUSS system planning, approach, and methodology currently in progress.

Discipline	Acronym	Description	Hardware
Integrated Design Tools Hardware	ICAD INTERGRAPH	Knowledge Based CAD System Knowledge Based CAD System	Symbolics (LISP) Intergraph
Expert System Tools	GEST	Generic Expert System Tool	Symbolics VAX 8600 (LISP)
Data Base Mgt Systems	RIM VBASE ORACLE	Relational Data Base Object Oriented Data Base Object Oriented Data Base	VAX Computer Vision, Ontologic IBM PC
Optimization	OPT SPARS FASTOP	Reduced Gradient Method Multispar Box Optimization Program Flutter and Strength Optimization Program	VAX, CYBER (FORTRAN)
Vehicle Synthesis	GASP VASCOMP II HESCOMP VSPEP AVID CDS SWEEP NAPSAP	General Aviation Sizing Program Tiltrotor Sizing and Performance Helicopter Sizing and Performance Vehicle Sizing & Perf Evaluation Program Aerospace Vehicle Interactive Program Configuration Development System Structural Weight Estimation Program Naval Airship Program for Sizing & Perf	VAX VAX, CYBER VAX, CYBER
Stability & Ctl	GTR MIMFOOF	Generalized Tiltrotor Program	VAX (FORTRAN)
Aeroelastic Stability	PASTA	Proprotor Aeroelastic Stability Program	VAX (FORTRAN)
Structural Analysis	NASTRAN ACCESS III APAS III MARC	NASA Structural Analysis Program Structural Analysis and Optimization Automated Program for A/C Structure	IBM (FORTRAN) VAX
Geometric Modelling	GEMPACK HESCAD	Aircraft Geometry Generator Helicopter Geometry Modeller	VAX (FORTRAN)
Finite Element Modelling	SACON	Expert System for MARC Analyzer	
Mission Analysis	GMAS	Goddard Mission Analysis System	
Aerodynamics	DYLOFLEX CLMAX USSAERO	Dynamic Loads of Flexible Airplanes Prediction of Aerodynamic Characteristics Aerodynamic Panel Loads Program	

Figure 12

Program Overview

An overview of the TRUSS program development plan is illustrated in Figure 13. The entire program is expected to run three-and-one-half years to final prototype demonstration. Major tasks consist of planning, system module development and integration, and frequent demonstrations to industry and government participants. Project planning will be fully addressed by the end of this year, with attention shifting to individual module management for the following years. TRUSS modules consist of the executive, relational and object-centered databases, user interface, geometric modelling facility, finite element modeller expert system, cost expert system, and SAGE (Structural Arrangement Generator and Evaluator). Currently in development, SAGE is an expert system responsible for structural concept synthesis and trade-off analyses, and will be discussed in more detail. All modules will be developed along parallel timelines, and integrated when they have reached sufficient maturity. It should be pointed out that at this time, participants for all of the modules have not been identified, but are expected to be within the next few months. Demonstrations include presentations on system concept formulation, computer demonstrations of TRUSS modules and their integration, module user validations, and finally a complete system validation.

Focus for LITE Research:
Tilt Rotor Unified Structural Design System

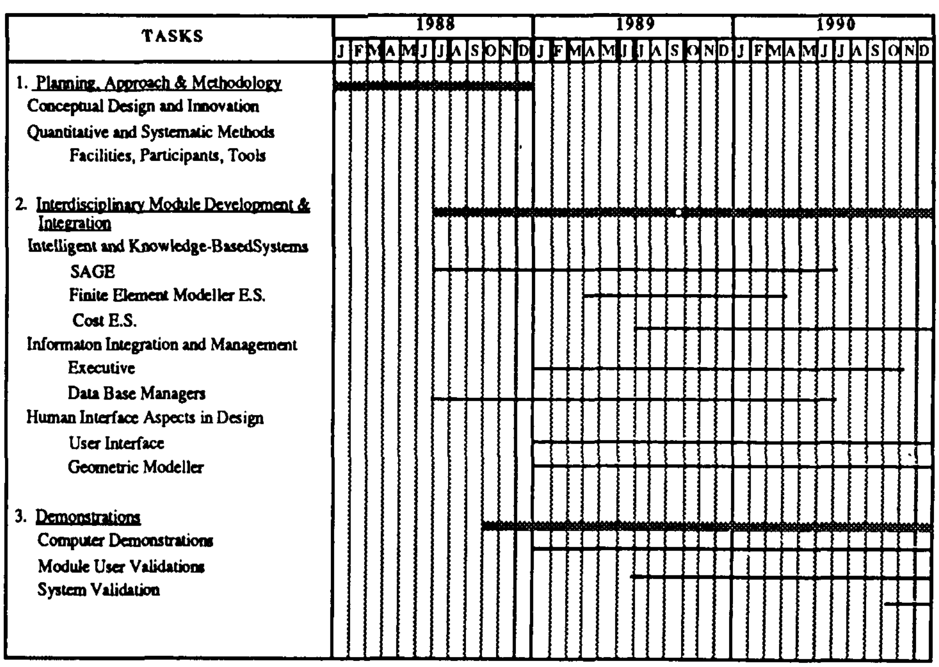


Figure 13

Description of SAGE

A man is referred to as a sage when he is known for his breadth of knowledge, wisdom, experience, and sound judgment. There is hardly a more appropriate acronym for the TRUSS module responsible for structural synthesis and trade-offs, an expert system called SAGE (Structural Arrangement Generator and Evaluator). Whereas there presently exist a number of excellent programs to analyze, size, and optimize structures, the next logical step in an automated design system is to capture the human design knowledge and expertise required to actually create and subjectively compare them.

Overview

Figure 14 shows a relational diagram for SAGE. Inputs to SAGE, in the form of structural design criteria, originate from the user and other conceptual analysis tools. They are directed by the executive, which stores them in appropriate databases. Wing geometry, fuel system specifications, weight and balance data, initial sizing criteria, and material specifications are some examples of these inputs. Using this information, the expert system may begin to configure a structural arrangement that satisfies most or all criteria. For example, the torque box cross sectional area for a two spar configuration can be roughly computed from torsional stiffness requirements and wing geometry. Then, the box problem may be decomposed into various design tasks, such as the front spar, rear spar, ribs, upper panel, and lower panel. These components may be synthesized individually, while keeping track of their functionality and interrelationships as a whole unit. Next, the structures for the box are sized via the executive using a variety of required modellers and analysis packages. Numerical and hybrid optimization techniques are employed to provide the structures with an equal basis for comparison. After all potential structures are sized, they may be evaluated heuristically with respect to level of reliability, maintainability, supportability, cost, and risk. Analytical methods in these areas are employed when possible. The results are feasible structures that have been automatically compared and appropriately ranked from best to worst concept. These may in turn be presented to the user in tabular or graphical form, from which detailed design may begin.

SAGE RELATIONAL DIAGRAM

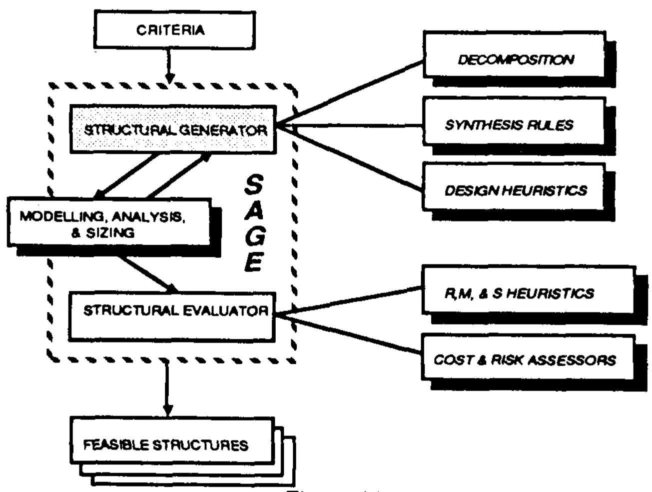


Figure 14

Reference 8 points out that the benefit of such automation is a reduction in the bottlenecks present in the current design process that arise from the unnecessary transfer of design information via humans. By efficiently closing the gaps in information exchange between humans and computers, using AI technology, much of the untimely engineering procedures present in complex system design that result in cost and time overruns can be remedied. With computers performing the more repetitive information tasks, human engineers will have the freedom to concentrate on more complex design problems.

From an academic standpoint, the development of SAGE is beneficial to research by providing a testbed for studying many fundamental design issues that can be applied not only to aircraft, but other complex transportation systems as well. Such issues include the nature of design, the sociocultural context of design, modelling the design process, design problem formulation, and the environment for design. Reference 9 discusses these in more detail.

One ongoing project of LITE similar to SAGE is MISSION (Reference 10), a knowledge-based system that explores the application of artificial intelligence to aircraft concept selection. The program selects one or more present technology aircraft to perform a given mission specified by the user. In addition, the system estimates initial sizing and performance characteristics for the different solution aircraft. MISSION currently has 23 different aircraft in its knowledge base, ranging from conventional fixed wing (supersonic and subsonic) configurations to conventional helicopters to hybrids like tiltrotors and vectored thrust concepts (Figure 15). The system is useful by providing the designer with a tool for rapid parametric studies of several entirely different aircraft types, and several different mission variations. More importantly for the development of SAGE, it serves as a useful testbed to new LITE members for understanding some of the issues involved in knowledge-based design systems, and gathering some proficiency in the use of an expert system building tool.

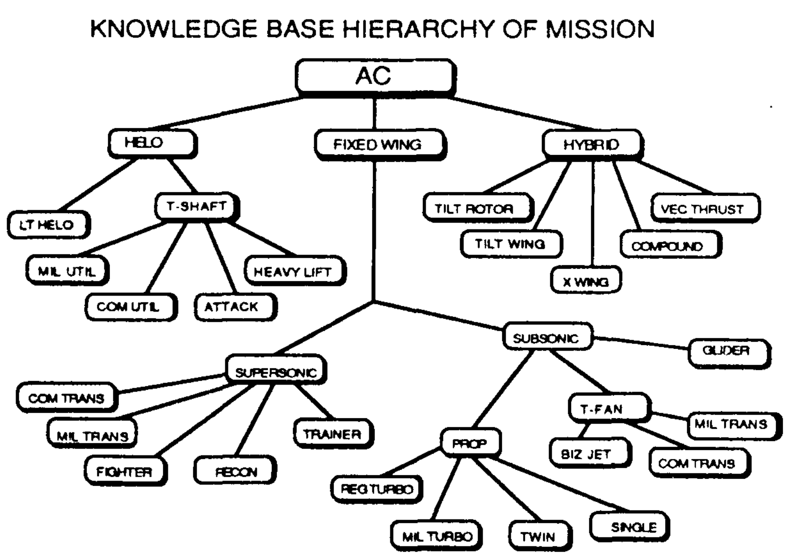


Figure 15

ORIGINAL PAGE IS OF POOR QUALITY

Expert System Tool

MISSION was developed using the Generic Expert System Tool (GEST), a product of the Georgia Tech Research Institute. Because of its features, availability, and hardware compatability, GEST was chosen for the development of SAGE. The software currently runs on all Symbolics, Texas Instruments TI Explorer, MicroVax II workstations, and VAX 8600 computers, providing a large variety of hardware to prototype the system and test integration with other modules of TRUSS. SAGE is being developed on the Explorer.

From Reference 11, GEST offers several advantageous features which make it one of the most comprehensive tools available today. Summarized in Figure 16, it has four knowledge representation schemes, including a production rule scheme which can generate hypotheses, deduce conclusions, and make additions and deletions to the design state. The inference engine supports three gears in all chaining modes: single steps through the rule base, continuous single rule firings, and continuous multiple firings. In addition, dynamic rule set modification and two levels of an explanation facility for the end-user alleviate the debugging task. Such features make GEST a robust tool that will expedite the development of SAGE.

GEST FEATURES

TOPIC	FEATURES	TOPIC	FEATURES
Knowledge Representation Schemes	FACTS FRAMES FUNCTIONS PRODUCTION RULE SCHEME	Inference Engine	FWD CHAIN BACKWARD BOTH 3 GEARS
Production Rule Scheme	HYPOTHESES CONCLUSIONS CHANGE DESIGN STATE	Rule Set Modification	EDITING CONFLICT DEPENDENCE SIMILARITY
Rule Conflict Resolution	COMPLEXITY RECENCY ANTECEDENT CONSEQUENT	Explanation Facility	PROGRAMMER USER
Evidential Reasoning	BAYESIAN SENSORY CERTAINTY FACTORS		

Figure 16

Architecture

Figure 17 shows the basic architecture of SAGE. First is the interface to the TRUSS executive, through which all data coming into or out of SAGE will pass. Next is the design state, or working memory, where pertinent design criteria, descriptions of candidate structural arrangements, and other local data reside. The remaining knowledge and rule bases are tied directly to the design state and to each other. The knowledge base contains domain specific knowledge for tiltrotor wings, required to understand the design state and act upon it. This includes a hierarchical breakdown of various structural arrangements used today (Figure 18), as well as knowledge of fasteners, basic pieces of structure, and how parts are related. The rule base drives the rest of the expert system, dynamically changing the design state, and making additions to the knowledge base as the design progresses.

Several categories of the rule base have been identified. When partial descriptions of the design criteria exist, a set of criteria rules attempts to complete them based on several sources, such as Federal Aviation Regulations. SAGE relies on the user to complete the criteria when it cannot. Procedural attachments are used to regulate any calls to outside programs that are needed to determine additional information for constraint generation and sizing.

Synthesis rules configure the major pieces of substructure in the system, such as the torque box and fixed trailing edge, and make sure that the interactions between them are accounted for. They will guide the configuration development in accordance with the structural design criteria. Shown in Figure 19, these rules model the design procedures used in industry today, which draw from a variety of sources such as company philosophy, FAR's, design manuals, and methods of manufacturing, maintenance, and support.

Evaluation rules will construct trade-off matrices that will be used to rank various choices of materials and structural concepts. After the structural concepts are optimally sized, they must be ranked according to some objective function, which may be cost, weight, level of maintainability, etc., or combinations thereof. SAGE will possess the knowledge required to evaluate the concepts both numerically and subjectively, presenting its results to the user for verification before the detailed design of structural subcomponents begins.

SAGE ARCHITECTURE

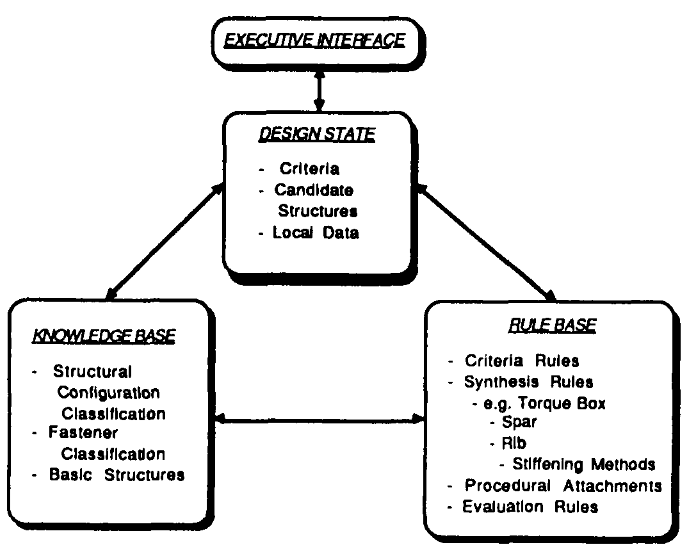


Figure 17

SAGE KNOWLEDGE BASE

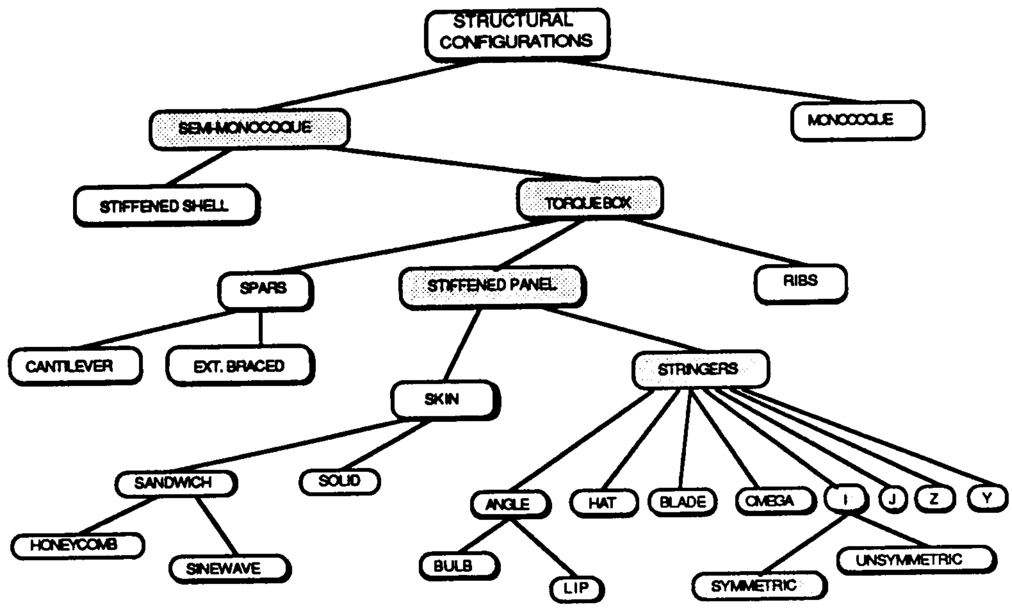


Figure 18

SYNTHESIS RULES

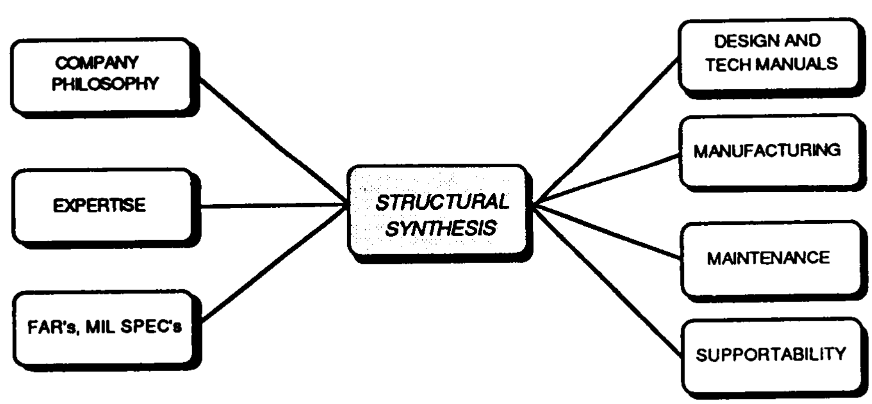


Figure 19

Timeline

The project schedule for SAGE is shown in figure 20. By the start of 1989, the problem scope will be narrowed to a specific task and formalized for all TRUSS participants. A three month phase of prototyping will commence, resulting in a basic working model of the expert system, which includes final rule and knowledge base prototypes, as well as local data management schemes. The majority of the remaining time will be spent on rule base modification and enhancement, accomplished by personal interviews with experts and development testing. User validation and completion of the prototype is expected in the first quarter of 1990.

SAGE OVERVIEW

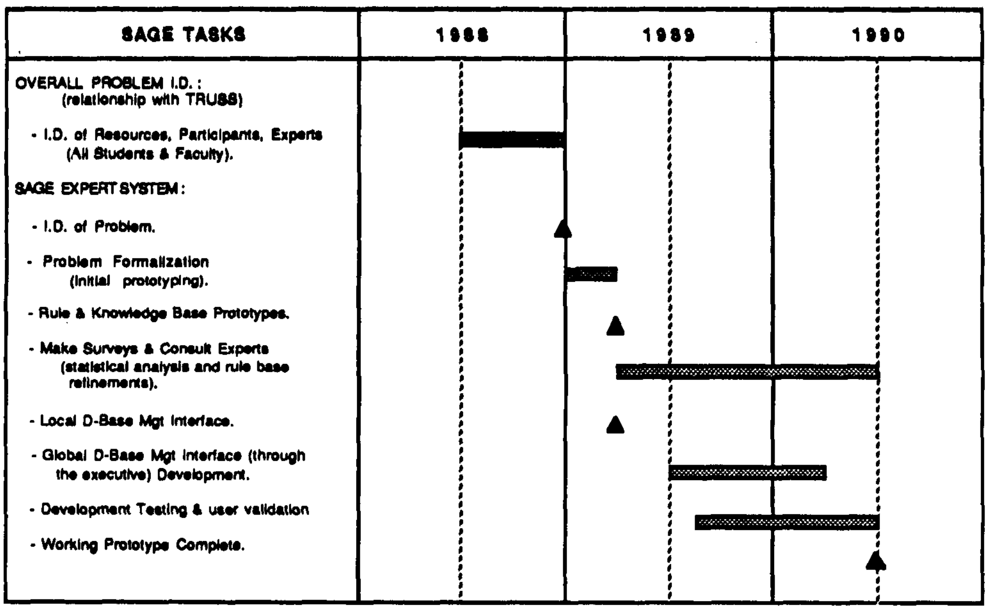


Figure 20

Example

This paper concludes with an example of how SAGE configures a wing structure. The first step is the examination of the overall wing characteristics. Figure 21 shows some typical specifications, giving the number of engines per wing, fuselage location, desired fuel system type, etc. SAGE searches its knowledge base for possible material types and makes selections based on their weight properties and compatibility with corrosion, for instance. Next, SAGE begins to arrange the basic pieces of tiltrotor wing structure: spars, ribs, panels, stringers, drive shafts, gearboxes, conversion spindle, pylon downstop, and so on. As it does so, analyses of lines of penetration check redundant load paths for damage tolerance. First level weights analysis ensures that the wing c.g. is within acceptable limits. Basic analysis of stiffness checks the aeroelastic stability merits of the pattern, and SAGE intelligently stiffens the structure with additional elements if necessary. Figure 22 shows how manufacturing methods are also considered by identifying necessary element spacing, as well as manufacturing access holes and closeouts.

All of this is done by combining the different pieces of structure in an intelligent manner, checking all of their feasible permutations. By doing so, the design space of a very large number of structural concepts is narrowed to a few in a systematic way.

When this process is finished, several versions of structural arrangements will exist and be specified like the one in Figure 21. These structures must now be modelled and sized. Geometric modelling and finite-element codes take the data from SAGE. These data include physical descriptions of the structural concepts, (honeycomb spars, integral hat-stiffened skin, etc.), geometric locations (fuselage station, wing station, relative geometries, etc.), and initial sizing data for the structural elements. From this point, other expert systems for finite-element modelling or drafting can use the data for their purposes. As the structures are sized, SAGE monitors the weights data, and makes adjustments to the structure if necessary. Iteration is required, and it may be found that some structures cannot be sized to meet the weight and stiffness requirements. SAGE flags these concepts and notifies the user, asking him to reject them or make changes to the design criteria.

Once the structures have been successfully sized, they are heuristically and numerically evaluated for levels of maintainability, reliability, supportability, cost, and risk. They are finally ranked and presented to the user tabularly and graphically as shown in Figure 23.

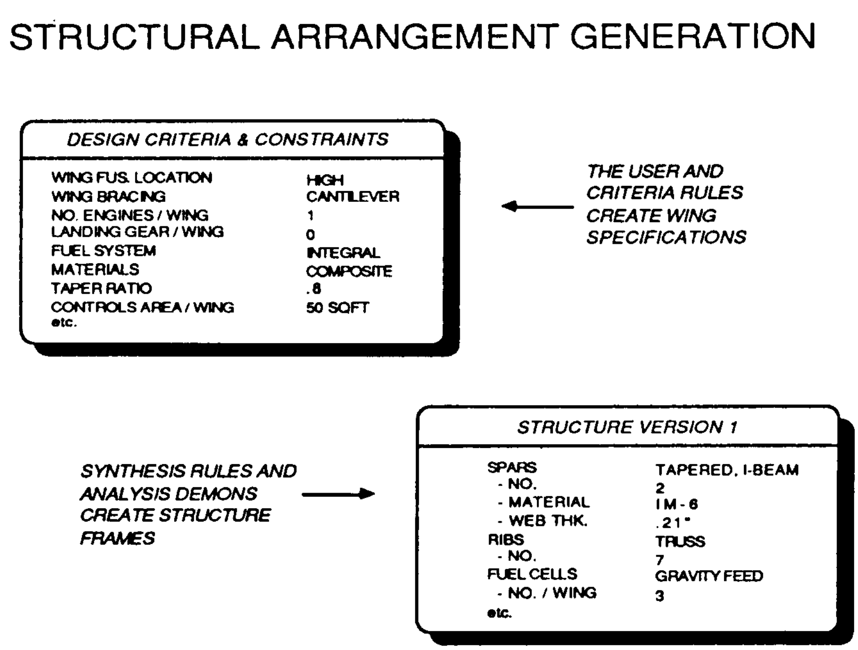


Figure 21

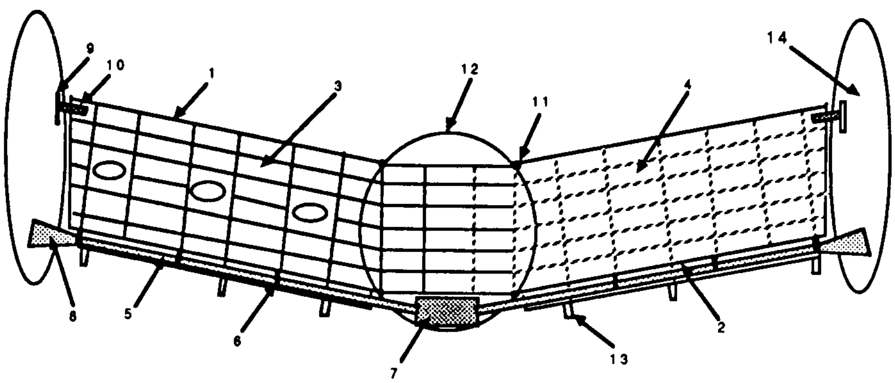
EXAMPLE

SYNTHESIS
RULE 100 : IF FUEL TANK SEALING IS INTEGRAL
THEN RIB SPACING IS AT LEAST 25 INCHES

EXPLANATION : In order to provide proper room for the assemblers to put sealant around ribs, spars, and other structural components comprising the fuel cell, a minimum of 25 inches between ribs is required.

Figure 22

TRUSS SYSTEM OUTPUT



1. Front Beam Assy
2. Rear Beam Assy
3. Lower Panel
4. Upper Panel
5. Drive Shaft
6. Drive Support Bearing
7. Center Gear Box
8. Conversion Spindle
9. Conversion Actuator
10. Conversion Actuator Spindle
11. Pin Fittings (4)
12. Fuselage Attachment Ring
13. Flaperon Hinge Fitting
14. Engine

Figure 23

Summary and Conclusions

This paper has reviewed TRUSS, an integrated design system that incorporates and addresses design issues currently persued by academia, government, and industry. While the program described is quite ambitious, it is the only way the proper focus can be given to this complex integration problem. The program will invoke faculty from several schools, as well as numerous graduate students. Close cooperation with industry is essential to obtain the required knowledge expertise. When the necessary hardware, software, and manpower are in place, significant research progress can rapidly be made in the field of complex systems design.

References

1. Johnson, V., "Minimizing Life Cycle Cost for Subsonic Commercial Aircraft," AIAA Paper 88-4402, 1988.

2. Kemp, A., "Effective Integration of Supportability Design Criteria into Computer Aided Design for the Conceptual Design Phase," AIAA Paper 88-4473, 1988.

3. "Goals and Priorities for Research in Engineering Design, a Report to the Design Research Community," The American Society of Mechanical Engineers, United Engineering Center, 345 East 47th St., New York, NY 10017.

4. Kroo, I., Takai, M., "A Quasi-Procedural, Knowledge-Based System for Aircraft Design," AIAA Paper 88-4428, 1988.

5. Britton, C., Jenkinson, W., "A Computer Aided Design System for Airplane Configuration," Computer Applications in Aircraft Design & Operation, Computational Mechanics Publications, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1987.

6. Staley, S., Anderson, D., "Functional Specification for CAD Databases," Computer-Aided Design Journal, 1986.

7. Dawson, C., "1993: A Vision of the Design Center," AIAA Paper 88-4451, 1988.

8. Hedenfels, R., "Integrated, Computer-Aided Design of Aircraft," Aircraft Design Integration & Optimization, Vol. I, AGARD Conference Proceedings No. 147, 1973.

9. Rouse, W., Boff, K., <u>System Design, Behavioral Perspectives on Designers, Tools, & Organizations</u>, Elsivier Science Publishing Co., Inc., New York, 1987.

10. Schrage, D., Bates, P., "The Configurator and Conceptual Design for Rotary Wing Aircraft," AIAA Paper 87-2891, 1987.

11. Gilmore, J., Ho, D., Howard, C., "GEST - The Generic Expert System Tool," Proceedings of the SPIE Applications of AI III, Vol. 635, 1986.